Amendments to the Claims:

A listing of the entire set of pending claims (including amendments to the claims, if any) is submitted herewith per 37 CFR 1.121. This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Previously presented) A method of audio encoding a stream that carries audio and video data, including:

encoding the audio data to provide a mean effective audio frame length $\overline{F}$ that equals a video frame length $1/f_V$ over an integral number M video frames,

wherein the encoding includes varying lengths *F* of the audio frames j in a defined sequence of frame lengths $F(j)$.

2. (Previously presented) The method of claim 1, wherein the frame length *F* is adjusted by varying an overlap *O* between successive audio frames.

3. (Previously presented) The method of claim 1 or claim 2, wherein the value $F(j)$ repeats periodically on *j*, the periodicity of $F(j)$ defining a sequence of frames.

4. (Previously presented) The method of claim 3 having M video and N audio frames per sequence, each audio frame being composed of *k* blocks of *t* samples each.

5. (Previously presented) The method of claim 4, wherein a total overlap O_T between frames in the sequence is equal to $O_T = p \times O + q \times (O+1)$, where *O* is an overlap length in blocks where $p \in \aleph \wedge q \in \aleph \wedge O \in \aleph \wedge O_T \in \aleph$.

6. (Previously presented) The method of claim 5, wherein only audio frames corresponding to a particular video frame are overlapped.

7. (Previously presented) The method of claim 6, wherein $p = (N - M) \times (O + 1) - O_T$ and $q = (N - M) - p$.

8. (Previously presented) The method of claim 5, wherein only audio frames corresponding to a particular video sequence are overlapped.

9. (Previously presented) The method of claim 8, wherein $p = (N - 1) \times (O + 1) - O_T$ and $q = (N - 1) - p$.

10. (Previously presented) The method of claim 5, wherein any adjacent audio frames are overlapped.

11. (Previously presented) The method of claim 10, wherein $p = N \times (O + 1) - O_T$ and $q = N - p$.

12. (Previously presented) The method of claim 4 in which $\exists n \in \aleph^+ : n \times t = M \times \left(\frac{f_A}{f_V} \right)$.

13. (Previously presented) A method of audio encoding a stream that encodes audio and video data, including

encoding audio samples of N quasi video-matched audio frames in frames with a defined sequence of overlap lengths,

wherein an effective length of the audio frames coincides with a length of a sequence of M video frames, where M and N are positive integers.

14. (Previously presented) A data stream encoded by the method of claim 13.

15. (Previously presented) The data stream of claim 14, wherein each of the audio frames is tagged to indicate a size of the audio frame.

16. (Previously presented) The data stream of claim 14, wherein each block of each audio frame is tagged to indicate whether or not the block is a redundant block.

17. (Previously presented) An audio encoder for coding audio for a stream that carries audio and video data, wherein the encoder produces audio frames of variable length such that a mean effective audio frame length $\overline{F}$ equals the video frame length $1/f_V$ over an integral number M video and N audio frames, and the audio frames j each have a variable overlap that provides an effective length *F* in a defined sequence of frame lengths $F(j)$ at encoding.

18. (Previously presented) The audio endcoder of claim 17, where the variable overlaps include a total of *p* short overlaps of length *O* and a total of *q* long overlaps of length *O*+ in an overlap sequence, the encoder calculating the overlap sequence using an algorithm that repeats after N frames.

19. (Previously presented) An audio decoder for decoding a stream that encodes audio and video data, wherein the decoder calculates an expected effective frame length of an incoming frame based on a defined sequence of frame lengths, adjusts the actual length of the incoming frame to make it equal to the expected frame length, determines whether any block within a received frame is a redundant block or a non-redundant block, mapping the non-redundant blocks onto sub-band samples.

20. (Previously presented) The audio decoder of claim 19, wherein the decoder is configured to modify the overlap status of blocks in the data stream by application of one or more of a set of block operators to each block.

21. (Previously presented) The audio decoder of claim 20, wherein the set of operators includes one or more of: NOP, an operator that does not change the status of a blocks; DROP, an operator that changes the first non-redundant block from the head overlap into a redundant block; APPEND, an operator that changes the first redundant block from the tail overlap into a non-redundant block; and SHIFT, an operator that is a combination of both DROP and APPEND operators.